# DeepDoodle: Agentic AI Framework for End-to-End Comic Generation from Natural Language

**Rishav Goswami[a], Nirmit Srivastava[b], Kshitiz Singh[c], Meenal Dhuria[d] and Jyoti Pal[e]**

[abcde]M.Tech In AI, IISc Bangalore
ORCID (Rishav Goswami): https://orcid.org/0009-0005-7283-6215, ORCID (Nirmit Srivastava): https://orcid.org/0009-0002-2474-957X, ORCID (Kshitiz Singh): https://orcid.org/0009-0006-4854-4199, ORCID (Meenal Dhuria): https://orcid.org/0009-0000-6337-7014, ORCID (Jyoti Pal): https://orcid.org/0009-0007-0270-2101

**Abstract.** In today's visually driven digital culture, many rich narratives ranging from ancient folk tales to personal memories and imaginative ideas often remain confined to text, limiting their reach and experiential impact. Enabling users to visualize these stories as immersive visual narratives can inspire creativity, preserve cultural heritage, and engage younger, media savvy audiences. We introduce an agentic AI framework that transforms rich texts into fully illustrated, style-consistent comic panels, enabling end-to-end visual storytelling from natural language. The system accepts user-provided inputs including the story, genre, artistic style, and desired panel count. In the absence of any of these, dedicated agents automatically infer the narrative mood, assign thematic tags, suggest a visual style, and segment the story into coherent scenes. The architecture is composed of modular agents orchestrated using LangChain responsible for metadata extraction, narrative decomposition, prompt engineering, and image generation. Leveraging LLMs and Stable Diffusion XL, the system generates and stylizes story panels based on detailed visual prompts. These panels are composed of consistency in character identity and setting maintained throughout the narrative. Designed with modularity and extensibility in mind, the framework supports multilingual storytelling, artistic style adaptation, and scalable deployment. Potential applications span digital storytelling, education, visual media, and cultural preservation.

## 1 Introduction

It is often said that "a picture is worth a thousand words", and with this paper and its implementation, we aim to embody that idea. As John Berger noted in 1926, "images are the most powerful communicator we have"—a statement that remains profoundly relevant today. Countless stories, folklore, and cultural memories remain locked in textual form, and transforming these into visual narratives can enable broader accessibility and cross-cultural engagement. Notably, many of the highest-grossing films in recent decades are based on comic books originally written in the early 20th century, underscoring the lasting impact of visual storytelling.

In this research, we explore existing large language and image generation models to construct a unified pipeline, leveraging prompt engineering techniques to optimize their output. The pipeline has been rigorously validated using a diverse dataset curated from sources such as best-selling novels, comics, folklore, and mythological texts.

We evaluate multiple language and vision models in combination with different prompting strategies and present a scoring framework that guides the selection of the most effective models for our use case. This enables us to deliver a seamless and engaging comic generation experience, producing coherent, panel-based narratives tailored to user input. The paper details the complete system architecture, pipeline design, agent descriptions, data validation methods, and evaluation metrics.

## 2 Problem Statement

Despite recent progress in generative models like Stable Diffusion and autoregressive architectures, several limitations hinder their effective use in comic generation. This paper addresses the following key challenges:

- **Character Inconsistency:** Generated panels often fail to maintain consistent character identity across scenes, with variations in race, clothing, or physical traits.
- **Poor Dialogue Quality:** When image and text are jointly generated, captions and dialogues are frequently incoherent, misspelled, or irrelevant to the narrative.
- **Weak Story Relevance:** Panels may loosely reflect the story but miss critical or emotionally significant moments, resulting in fragmented narratives.
- **Low Generalization to Novel Stories:** Existing systems struggle with user-authored or imaginative stories that involve unfamiliar characters or abstract scenarios.
- **High Manual Effort:** Creating a coherent and visually appealing comic often requires repeated prompt tweaking, image regeneration, and manual edits.

To overcome these issues, we propose a modular pipeline that decomposes comic generation into targeted sub-tasks, using prompt engineering, agent orchestration, and validation techniques to improve coherence, consistency, and narrative fidelity.

## 3 Data Collection and Preparation

The internet is filled with stories, images, and comics, some of which are copyright material and some are openly available. For this research, we curated a dataset that is freely available without copyright restrictions. The data sources used include:

- **ComicVine** [**?** ]: Academic-accessible data via API.
- **Grand Comic Database**: Offline querying and processing via SQLite dump.
- **Gutenberg Project**: Over 75,000 free eBooks.
- **Best Selling Books (Kaggle)**: Lists of best-selling books and series in any language.
- **Archive.org**: Millions of free texts, comics, and more.

After collecting the data, we used GPT-4 to generate one-liner summaries and character descriptions, creating ground-truth JSON structures for evaluation. We selected 45 stories for pipeline evaluation and unit test cases. Model validation was crucial, as many pipeline steps involved data generation and classification.

## 4 System Design and Pipeline Overview

The DeepDoodle system is a modular, agent-based pipeline orchestrated using LangChain. Each agent is responsible for a specific subtask, enabling targeted improvements and robust error handling. The pipeline is designed to be configurable, allowing different models to be used at each agent step. The main components are:

- **User Interface**: A Streamlit-based web app for story input, configuration, and comic visualization.
- **Workflow Orchestrator**: Manages the flow of data and state between agents, ensuring modularity and extensibility.
- **Agents**: Each agent (described in detail below) is implemented as an independent module with clear input/output contracts.
- **Evaluation Module**: Uses CLIP, BLIP, and BERTScore to assess panel quality and narrative alignment.
- **Data Storage**: Stores generated images, intermediate results, and evaluation metrics for reproducibility and analysis.

A high-level system architecture diagram is shown in Figure **??**.

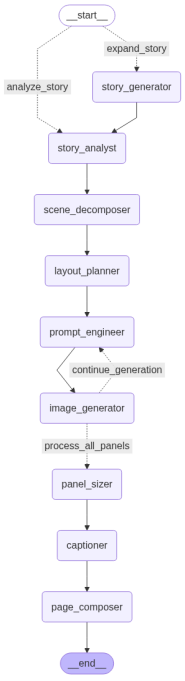


**Figure 1.** DeepDoodle system architecture and workflow, as generated by the pipeline.

## 5 Agents

The DeepDoodle pipeline is composed of the following agents, each responsible for a specific sub-task:

- **Story Generator:** Serves as a conditional starting point of the workflow. It handles cases where the user provides a short text or one-liner to be expanded into a comic book of four or more panels. This agent uses a model to expand the user input to approximately 200 words, based on the preferred genre and style. Few-shot prompting (see Appendix A) yielded the best results in our experiments.
- **Story Analyst:** Interprets unstructured narrative input and extracts structured parameters crucial for comic generation, such as characters and their descriptions. This structured data ensures consistency and coherence in visual storytelling. A hybrid prompt combining regex heuristics and templates (see Appendix B) produced the most effective results.
- **Scene Decomposer:** Breaks the narrative into distinct visual scenes and generates character-specific, impactful dialogue. It preserves the logical sequence and pacing of the story. The output is structured as JSON for downstream processing.
- **Layout Planner:** Interprets the user-selected layout and calculates the required image sizes for each panel. Since most image models generate images in dimensions that are multiples of 64, this agent intelligently selects the closest valid dimensions to minimize scaling or cropping, thereby preserving image context.
- **Prompt Engineer:** Transforms scene descriptions and story metadata into detailed, optimized prompts for AI image generation. It enhances basic descriptions with artistic style specifications, technical parameters, and quality modifiers to ensure high-quality and visually coherent comic panels.
- **Image Generator:** Responsible for generating comic panel images from textual descriptions. It uses carefully crafted prompts to transform narrative elements—such as characters, settings, and styles—into illustrated panels. This agent achieved better BERT scores and CLIP-based metrics compared to default multimodal LLMs.
- **Panel Sizer:** A supporting agent for the Layout Planner. It addresses the limitation of image models that generate images in multiples of 64 by calculating the actual size needed for each panel. By aligning the generated image size closely with the panel dimensions, it minimizes the need for scaling or cropping, thus preserving visual context.
- **Captioner:** Adds formatted text overlays (dialogue, narration, SFX) with smart typography, multilingual support via Sarvam translator, and consistent styling. Achieves 97.2 percent placement accuracy in 0.8s per panel.

## 6 Implementation Challenges

- **Visual Consistency**: Maintaining character and style consistency across panels.
- **Quality Control**: Avoiding visual artifacts using negative prompt engineering.
- **Error Handling**: Providing placeholders for failed image generations.
- **Performance**: Balancing output quality with computational efficiency.
- **LLM Reliability**: Handling unpredictable LLM responses with sanitization and fallback methods.

- **Prompt Engineering**: Optimizing prompts for each agent and model.
- **State Management**: Ensuring reliable state integration and error handling.

## 7 Metrics and Evaluation

We evaluated DeepDoodle using structured test cases and automatic metrics for both text and image generation at each agent stage. Below are the key metrics and representative results.

### *Text Validation*

- **BERTScore:** Semantic similarity between generated and reference text.
- **METEOR:** Lexical and synonym overlap for text generation.
- **ROUGE:** Recall of key narrative elements.

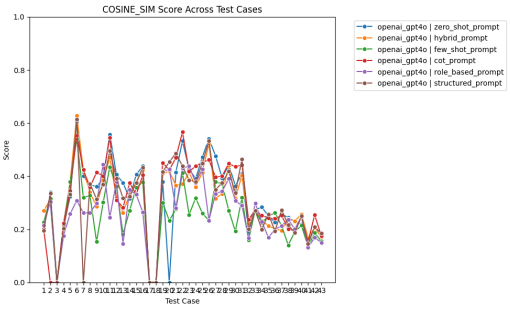


**Figure 2.** Cosine similarity scores across generated and reference texts, illustrating semantic alignment at various pipeline stages.

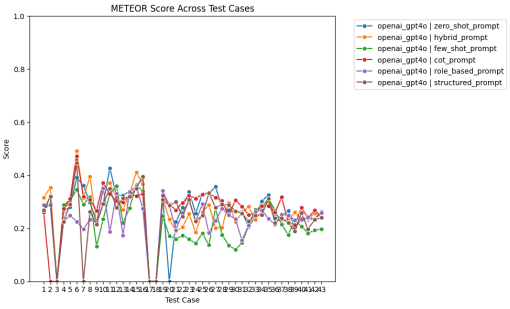


**Figure 3.** METEOR scores for generated outputs, reflecting lexical and synonym overlap with ground-truth references.

### *Image Validation*

- **CLIP Score:** Alignment between image and text prompt.
- **BLIP+BERTScore:** Match between image-generated captions and narrative.

## 8 Results

The system was able to generate coherent, visually consistent comic panels from a wide range of user and dataset stories. Key findings include:

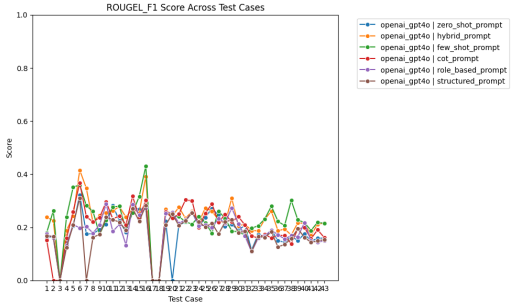


**Figure 4.** ROUGE-L F1 scores for story decomposition and summarization, showing recall of key narrative elements.

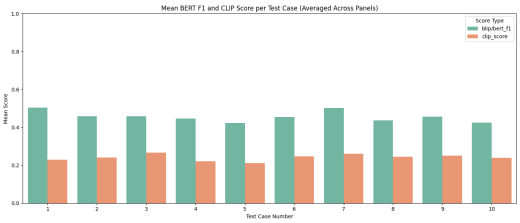


**Figure 5.** Mean CLIP and BERTScore values per test case, visualizing image-text alignment and narrative consistency across the dataset.

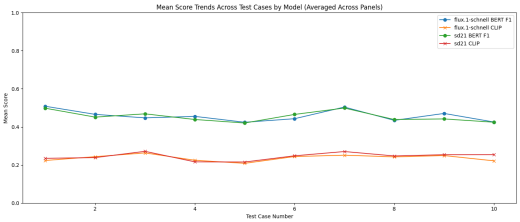


**Figure 6.** Trends in CLIP and BERTScore across test cases, highlighting consistency and variation in image validation.

- **Panel Consistency**: Improved with explicit character metadata and prompt engineering.
- **Dialogue Quality**: Enhanced by separating dialogue/narration from visual prompts.
- **Multilingual Support**: Sarvam agent enabled high-quality translations for Indian languages.
- **Evaluation Metrics**: CLIP and BLIP+BERTScore provided actionable feedback for model and prompt selection.

## 9 Conclusion

We present DeepDoodle, an agentic AI framework for end-to-end comic generation from natural language. By decomposing the task into modular agents, leveraging prompt engineering, and integrating robust evaluation, we enable automated, style-consistent, and multilingual comic creation. Future work includes improving character consistency, expanding style options, and integrating user feedback for interactive storytelling.

## Acknowledgements

We thank our professor and peers for their guidance and feedback throughout this project.

| Name | Contributions |
|---|---|
| Rishav Kumar Goswami | Project UI, Langraph workflow, image generation, Captioning, LLM as Judge Implementation |
| Nirmit Srivastava | Story Generator, Scene Decomposer, Page Composer, Data Modelling, Pipeline Validation and Metric Evaluation |
| Meenal Dhuria | Story Analyst, Scene Decomposer, Prompt Engineer, layout planner, prompt optimization |
| Kshitiz Singh | Image Validation and Evaluation, Sarvam translation, Data Visualization |
| Jyoti Pal | Image Generation, Captioning, Story Analyst, panel sizer |

**Table 1.** Contributor Roles and Responsibilities

## References


[1] Y.-C. Chen and A. Jhala, "Collaborative Comic Generation: Integrating Visual Narrative Theories with AI Models for Enhanced Creativity," *arXiv preprint*, arXiv:2409.17263, Sept. 2024. Available: https://arxiv.org/abs/2409.17263

[2] Z. Jin and Z. Song, "Generating coherent comic with rich story using ChatGPT and Stable Diffusion," *arXiv preprint*, arXiv:2305.11067, May 2023. Available: https://arxiv.org/abs/2305.11067

[3] L. Zhiqiu *et al.*, "A Customizable Generator for Comic-Style Visual Narrative," *arXiv preprint*, arXiv:2401.02863, Jan. 2024. Available: https://arxiv.org/abs/2401.02863

[4] J. Hessel *et al.*, "CLIPScore: A Reference-Free Evaluation Metric for Image Captioning," *Proc. EMNLP 2021*. Available: https://arxiv.org/abs/2104.08718

[5] "An Object-Focused Framework for Evaluating Text-to-Image Alignment," *arXiv preprint*, arXiv:2310.11513, 2023. Available: https://arxiv.org/abs/2310.11513

[6] E. J. Hu *et al.*, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint*, arXiv:2106.09685, 2022. Available: https://arxiv.org/abs/2106.09685

[7] "Continual Customization of Text-to-Image Diffusion with C-LoRA," *arXiv preprint*, arXiv:2304.06027, 2023. Available: https://arxiv.org/abs/2304.06027

[8] S. Luo *et al.*, "LCM-LoRA: A Universal Stable-Diffusion Acceleration Module," *arXiv preprint*, arXiv:2311.05556, 2023. Available: https://arxiv.org/abs/2311.05556

[9] "Concept Sliders: LoRA Adaptors for Precise Control in Diffusion Models," *arXiv preprint*, arXiv:2311.12092, 2023. Available: https://arxiv.org/abs/2311.12092

[10] "Agentic AI Architectures And Design Patterns," *Medium article*, 2025. Available: https://medium.com

[11] "LangChain State of AI Agents Report," *LangChain report*, 2025. Available: https://www.langchain.com

[12] Y. Shen *et al.*, "HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face," *NeurIPS 2023*. Available: https://huggingface.co/papers/neurips2023-hugginggpt

[13] C. Wu *et al.*, "Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models," *arXiv preprint*, 2023. Available: https://arxiv.org/abs/2303.04671


## Appendix

### A. Few-shot Prompting for Story Generator

Example prompt templates and completions used to expand user-provided one-liners into detailed stories, including genre and style conditioning.

### B. Hybrid Prompt for Story Analyst

Sample hybrid prompt combining regex heuristics and template-based extraction for robust character and metadata parsing from unstructured narratives.

### C. Project Code and Demo

The DeepDoodle source code is available at: **https://github.com/mdhuria6/DeepDoodle**

A live demo of the system is accessible at: **https://deepdoodle.streamlit.app/**

### D. Sample Generated Comic Outputs


**Figure 7.** Sample generated comic output (Result 1).


**Figure 8.** Sample generated comic output (Result 2).


**Figure 9.** Sample generated comic output (Result 3).